**Funcionamiento:**

El funcionamiento de la aplicación ClientesAgenciaDeViajes basada en el patrón de arquitectura **MVC**  se puede explicar de la siguiente manera:

1. Para hacer una búsqueda el usuario interactúa con la interfaz grafica GUIBusquedaClientes, selecciona una profesión y da clic a buscar
2. GUIBusquedaClientes notifica a al controlador GUIBusquedaClientesController este a su vez gestiona el evento.
3. El controller accede a modelo para ejecutar el método buscarClientesPorProfesion busca los clientes de una profesión, cambia el estado del objeto y notifica a los observadores.
4. Las vistas actualizan su información.

Ventajas y Desventajas de la arquitectura **MVC:**

**Se tienen muchas ventajas como:**

* La implementación se realiza de forma modular.
* Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
* Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
* Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
* MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

**Como desventajas tenemos:**

* Para desarrollar una aplicación bajo el patrón de diseño MVC es necesario una mayor dedicación en los tiempos iníciales del desarrollo. Normalmente el patrón exige al programador desarrollar un mayor número de clases que, en otros entornos de desarrollo, no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es mucho más mantenible, extensible y modificable que una aplicación que no lo implementa.
* MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación. Esta arquitectura inicial debe incluir, por lo menos, un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.
* MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.